

Amy Miles

**SYSTEM, METHOD AND ARTICLE OF MANUFACTURE TO FACILITATE
REMOTE STATION ADVERTISING**

5

Reference to Pending Applications

This application is a continuation-in-part application based on Provisional Patent Application 60/200,483 filed on April 28, 2000 and entitled "System, Method and Article of Manufacture To Facilitate Fuel Station Advertising".

10

Reference to Microfiche Appendix

This application is not referenced in any microfiche appendix.

15
20
25
30
35
40
45
50
55
60
65
70
75
80
85
90
95
100
105
110
115
120
125
130
135
140
145
150
155
160
165
170
175
180
185
190
195
200
205
210
215
220
225
230
235
240
245
250
255
260
265
270
275
280
285
290
295
300
305
310
315
320
325
330
335
340
345
350
355
360
365
370
375
380
385
390
395
400
405
410
415
420
425
430
435
440
445
450
455
460
465
470
475
480
485
490
495
500
505
510
515
520
525
530
535
540
545
550
555
560
565
570
575
580
585
590
595
600
605
610
615
620
625
630
635
640
645
650
655
660
665
670
675
680
685
690
695
700
705
710
715
720
725
730
735
740
745
750
755
760
765
770
775
780
785
790
795
800
805
810
815
820
825
830
835
840
845
850
855
860
865
870
875
880
885
890
895
900
905
910
915
920
925
930
935
940
945
950
955
960
965
970
975
980
985
990
995

Technical Field of the Invention

In general, the present application relates to automated advertising displays. In particular, the present invention relates to a system, method and article of manufacture for retrieving, dynamically modifying and presenting audibly and visually perceptible content upon a remote presentation device.

20

Background of the Invention

Advertising systems and display units intended for use in remote locales are well represented in the prior art. For example:

United States Patent No. 5,134,716 issued on July 28, 1992 to David J. Craig, subsequently assigned to Caltex Oil Pty. Limited discloses a Point of Sale Audio-Visual Advertising System which has a central station and a plurality of outstations. The central station is generally located in

a shop attached to service or filling stations which sell petroleum products such as gasoline, while the outstations are located at self-service pumps located on the driveway of the service station. The system provides audio-visual advertising material to the purchaser while the tank filling operation is in progress and immediately prior to his entry into the shop area.

5 United States Patent No. 5,717,374 issued on February 10, 1998 to Harry F. Smith and United States Patent No. 5,914,654 issued on June 22, 1999 to Harry F. Smith, both subsequently assigned to Intellectual Property Development Associates of Connecticut, Inc., discloses a Methods and Apparatus For Inputting Messages, Including Advertisements, To A Vehicle. The methods and apparatus are disclosed for inputting messages and other information, such as advertisements, to a vehicle while the vehicle is coupled to a local station, such as a recharging station or a refueling station. The messages can be selected in accordance with information received from the vehicle, including information that selectively identifies one, some or all of: (a) a characteristic of an occupant of the vehicle (e.g. name, account number, address, etc.); (b) a characteristic of the vehicle (e.g. make, model, year, class, registration number, marker number, odometer reading, owner, etc.); (c) a destination of the vehicle (entered through a data entry console and optionally stored within a vehicle memory); and (d) any other characteristic of interest.

10
15
20 United States Patent No. 5,980,090 issued on November 9, 1999 to William C. Royal Jr. and Randall O. Watkins, subsequently assigned to Gilbarco, Inc., discloses an Internet Asset Management System For A Fuel Dispensing Environment. This invention provides communication servers at each device in a fueling environment and connecting the servers to a common network. The network may be a local network or a largely remote network, such as the Internet. Preferably, in either embodiment, primary communications between these devices and any devices accessible

via the Internet use the hypertext transfer protocol (HTTP) and hypertext markup language (HTML).

In particular, each device server is adapted to facilitate real-time access between the device server and the remote device upon access of a particular page, script or function. In particular, the present invention relates to embedding executable content onto an HTML page so that when the page is loaded into an HTML browser after being accessed, the executable content starts running automatically.

United States Patent No. 6,032,126 issued on February 29, 2000 to David L. Kaehler, subsequently assigned to Gilbarco, Inc., discloses an Audio and Audio/Video Operator Intercom For A Fuel Dispenser. An apparatus for installation in a retail setting for selling fuel and other products ordered by a customer interacting with an operator. A video control system interfaces an external audio/video signal source with an audio/video signal source from an other product ordering apparatus operator. The external audio/video source transmits advertising and promotional materials to a video display located on a card reader equipped fuel dispenser. Additionally, customers can signal and communicate through audio/video signals with the operator to order other merchandise. Total transaction costs for fuel and non-fuel products is provided and paid for at the fuel dispenser.

However, none of the afore cited references clearly delineate, discuss, disclose or claim a system, method or article of manufacture whereby timely geographically relevant information can be purposed and presented to specific refueling locals in conjunction with or independent of advertising content also intended for the specific local.

Brief Summary of the Invention

An audio/visual advertising system is disclosed which has one or more computer compatible communication networks content acquisition and display presentation central processing units, multiple function presentation units and first and second software means which allow for the retrieval and presentation of advertising and geographically relevant content.

Consequently, it is an objective of the instant invention to allow for the acquisition, dispersion and presentation of geographically relevant information to a selective receiving location within a larger universe of such locations.

It is a further objective of the instant invention to allow for the presentation of advertising media at a remote station accompanied with geographically relevant information, such as but not limited to, weather, weather alerts, sporting news, national or regional news, etc.

It is another objective of the instant invention to allow the consumer to interact with the present invention via a touch screen communications capability.

Another objective of the instant invention is to allow the consumer to interact with the present invention via an audibly receptive communications capability.

It is another objective of the instant invention to provide for the presentation of geographically relevant information and advertising media at a remote station utilizing a computer compatible communication means, such as computerized networks including terrestrial and satellite communications.

Another objective of the instant invention is to allow for a first transmission of advertising media supplemented with subsequent repeated transmissions providing updated geographically relevant information within a predetermined time interval.

Description of the Drawings

Figure 1 is a system schematic illustrating the invention's primary hardware and software processing components as practiced in its preferred embodiment.

Figure 2 illustrates a program component which facilitates the retrieval, processing and transmitting of ticker information.

Figure 3 illustrates the programming means steps which facilitate the processing and transmitting of image files.

Figure 4 is a flowchart representing Phase II software functionality.

Figure 5 is a simplified diagram denoting primary processing steps associated with data acquisition as practiced in the invention's preferred embodiment.

Figure 6 is a simplified diagram denoting primary processing steps associated with fueling site presentation of acquired data and advertising content as practiced in the invention's preferred embodiment.

Figure 7 is an illustrative information presentation format as practiced by the invention in its preferred embodiment.

Detailed Description of the Preferred Embodiment

While the making and using of various embodiments of the present invention are discussed in detail below, it should be appreciated that the present invention provides for inventive concepts capable of being embodied in a variety of specific contexts. The specific embodiments discussed herein are merely illustrative of specific manners in which to make and use the invention and are not to be interpreted as limiting the scope of the instant invention.

The claims and the specification describe the invention presented and the terms that are employed in the claims draw their meaning from the use of such terms in the specification. The same terms employed in the prior art may be broader in meaning than specifically employed herein. Whenever there is a question between the broader definition of such terms used in the prior art and the more specific use of the terms herein, the more specific meaning is meant.

While the invention has been described with a certain degree of particularity, it is clear that many changes may be made in the details of construction and the arrangement of components without departing from the spirit and scope of this disclosure. It is understood that the invention is not limited to the embodiments set forth herein for purposes of exemplification, but is to be limited only by the scope of the attached claim or claims, including the full range of equivalency to which each element thereof is entitled.

Figure 1 is a system schematic illustrating primary hardware and software components of the instant invention as practiced in its preferred embodiment. In Figure 1, first software 3 executing within a data acquisition central processing unit 5 facilitates the acquisition of advertising and geographically relevant data. The present invention allows for such data acquisition to be facilitated via a plurality of means. For instance, data may be presented to the data acquisition central

processing unit 5, without limitation, via a transporable storage means, such as a floppy disk or computer readable recording tape commonly used to distribute advertising content or compact disk. Additionally, said first software 3 can acquire content via a computer compatible network communication means. Examples of such computer compatible network communication means well known and practiced in the art would include, without limitation, local area networks (LAN's), wide area networks (WAN's) metropolitan area networks (MAN's), campus area networks (CAN's), Extranets, Intranets and the Internet. Content so acquired will typically be represented as geographically relevant information with multiple and diverse geographic records delineated via the insertion of a delineation code appended to a relevant record or records incorporated within an acquisition file. Said appending of said delineation code is facilitated by said first software 3. The delineated information is then compiled into a computer recognizable file structure, and stored to a storage means accessible to said data acquisition central processing unit 5. In the invention's preferred embodiment said storage means is represented and accessible as a page associated with a Web-site, though any file structure recognized by a computer based transmission protocol will facilitate practice of the invention. In the invention's preferred embodiment, File Transfer Protocol (hereinafter, referred to synonymously referred to as "FTP") and M-PEG file structures are utilized.

Continuing with discussion relating to Figure 1. The stored file embodying geographically delineated records is then transmitted over a communications link 7 to a transceiver facility, such as, but not limited to a satellite office central processing unit 10. Once resident and accessible to the transceiver facility 10, the transceiving facility 10 further transmits the received file to a satellite transceiver 15 over a second communications link, or hard wired connection 12 for subsequent transmission to an orbiting satellite 20, via a third communications link 16. The orbiting satellite

20 then transmits or otherwise broadcasts the delineated file to a plurality of geographically dispersed terrestrial receiving units 24, via a fourth communications link 22. Said geographically dispersed terrestrial receiving units 24 possess a receiving decoding means by which information geographically relevant to the physical location of said the ground unit 24 is decoded and presented to a remote site central processing unit 25. Such selective receipt decoding and presentation of said information would be directed towards as an example, include retail locations such as retail outlets within a specific neighborhood, town, city, state, region, or country. Said means of presenting selective content to local central processing unit(s) from a selective decoding device are well known and practiced by those skilled in the art. As an alternative communication means to facilitate the transfer of geographically relevant information and advertising media between said data acquisition central processing unit 5 and said remote site central processing unit 25, a computer compatible network such as the Internet may be used as an alternative to augment, supplement or replace the aforestated satellite base communication process. Second software 26 executing within the remote site central processing unit 25 then identifies and divides audio, video and text records contained within the transmitted file intended for received ground unit 24 and, stores such information in separate and distinct file structures accessible to said remote site central processing unit 25 and transmits said previously acquired content to an information presentation device 29. Said transmission occurring and facilitated via radio frequency (RF) transmission/receiving means 27 well known and practiced by those skilled in the art. Alternatively, said transmission may be facilitated via a hard wire/cable communication link 27 between said remote site central processing unit 25 and said information presentation device 29.

Software processes of the present invention as practiced in its preferred embodiment can best be understood as occurring in two distinct phases. Phase I (a.k.a "first software") 3, executed within a data acquisition central processing unit 5, and Phase II (a.k.a. "second software") 26, executed from within a remote site central processing unit 25 . Phase I of source code for said first software is immediately included herein for purposes of providing full and enabling disclosure.

```
' Reg Key Security Options...
Const KEY_ALL_ACCESS = &H2003F
```

```
' Reg Key ROOT Types...
Const HKEY_LOCAL_MACHINE = &H80000002
Const ERROR_SUCCESS = 0
Const REG_SZ = 1           ' Unicode nul terminated string
Const REG_DWORD = 4        ' 32-bit number
```

```
Const gREGKEYSYSINFOLOC = "SOFTWARE\Microsoft\Shared Tools Location"
Const gREGVALSYSINFOLOC = "MSINFO"
Const gREGKEYSYSINFO = "SOFTWARE\Microsoft\Shared Tools\MSINFO"
Const gREGVALSYSINFO = "PATH"
```

```
Private Declare Function RegOpenKeyEx Lib "advapi32" Alias "RegOpenKeyExA" (ByVal hKey As Long, ByVal lpSubKey As String, ByVal ulOptions As Long, ByVal samDesired As Long, ByRef phkResult As Long) As Long
Private Declare Function RegQueryValueEx Lib "advapi32" Alias "RegQueryValueExA" (ByVal hKey As Long, ByVal lpValueName As String, ByVal lpReserved As Long, ByRef lpType As Long, ByVal lpData As String, ByRef lpcbData As Long) As Long
```

```
Private Declare Function RegCloseKey Lib "advapi32" (ByVal hKey As Long) As Long
```

```
Private Sub Form_Load()
    LoadResStrings Me
    lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
    lblTitle.Caption = App.Title
End Sub
```

```
Private Sub cmdSysInfo_Click()
    Call StartSysInfo
End Sub
```

```
Private Sub cmdOK_Click()
    Unload Me
End Sub
```

```
Public Sub StartSysInfo()
    On Error GoTo SysInfoErr
```

```
    Dim rc As Long
    Dim SysInfoPath As String
```

```

' Try To Get System Info Program Path\Name From Registry...
If GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFO, gREGVALSYSINFO, SysInfoPath) Then
' Try To Get System Info Program Path Only From Registry...
    Elseif GetKeyValue(HKEY_LOCAL_MACHINE, gREGKEYSYSINFOLOC, gREGVALSYSINFOLOC,
5 SysInfoPath) Then
        ' Validate Existence Of Known 32 Bit File Version
        If (Dir(SysInfoPath & "\MSINFO32.EXE") <> "") Then
            SysInfoPath = SysInfoPath & "\MSINFO32.EXE"

10         ' Error - File Can Not Be Found...
        Else
            GoTo SysInfoErr
        End If
    ' Error - Registry Entry Can Not Be Found...
15 Else
        GoTo SysInfoErr
    End If

    Call Shell(SysInfoPath, vbNormalFocus)

20 Exit Sub
SysInfoErr:
    MsgBox "System Information Is Unavailable At This Time", vbOKOnly
End Sub

25 Public Function GetKeyValue(KeyRoot As Long, KeyName As String, SubKeyRef As String, ByRef KeyVal As String)
    As Boolean
        Dim i As Long                ' Loop Counter
        Dim rc As Long                ' Return Code
        Dim hKey As Long              ' Handle To An Open Registry Key
        Dim hDepth As Long            '
        Dim KeyValType As Long        ' Data Type Of A Registry Key
        Dim tmpVal As String           ' Temporary Storage For A Registry Key Value
        Dim KeyValSize As Long        ' Size Of Registry Key Variable
35 '-----
    ' Open RegKey Under KeyRoot {HKEY_LOCAL_MACHINE...}
    '-----
    rc = RegOpenKeyEx(KeyRoot, KeyName, 0, KEY_ALL_ACCESS, hKey) ' Open Registry Key

40 If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Error...

    tmpVal = String$(1024, 0)          ' Allocate Variable Space
    KeyValSize = 1024                  ' Mark Variable Size

45 '-----
    ' Retrieve Registry Key Value...
    '-----
    rc = RegQueryValueEx(hKey, SubKeyRef, 0, KeyValType, tmpVal, KeyValSize) ' Get/Create Key Value

50 If (rc <> ERROR_SUCCESS) Then GoTo GetKeyError    ' Handle Errors

    tmpVal = VBA.Left(tmpVal, InStr(tmpVal, VBA.Chr(0)) - 1)
    '-----
    ' Determine Key Value Type For Conversion...
55 '-----

```

```

        Select Case KeyValType
        Case REG_SZ
            KeyVal = tmpVal
        Case REG_DWORD
            Data Type
                For i = Len(tmpVal) To 1 Step -1
                    KeyVal = KeyVal + Hex(Asc(Mid(tmpVal, i, 1)))
                Next
                KeyVal = Format$("&h" + KeyVal)
            End Select

            GetKeyValue = True
            rc = RegCloseKey(hKey)
            Exit Function

        ' Search Data Types...
        ' String Registry Key Data Type
        ' Copy String Value
        ' Double Word Registry Key
        ' Convert Each Bit
        ' Build Value Char. By Char.
        ' Convert Double Word To String

        ' Return Success
        ' Close Registry Key
        ' Exit

    GetKeyError: ' Cleanup After An Error Has Occured...
        KeyVal = ""
        GetKeyValue = False
        rc = RegCloseKey(hKey)

        ' Set Return Val To Empty String
        ' Return Failure
        ' Close Registry Key

    End Function

VERSION 5.00
Begin VB.Form frmAbout
    BorderStyle = 3 'Fixed Dialog
    Caption = "About MediaVision"
    ClientHeight = 3630
    ClientLeft = 45
    ClientTop = 330
    ClientWidth = 5865
    ClipControls = 0 'False
    LinkTopic = "Form1"
    MaxButton = 0 'False
    MinButton = 0 'False
    ScaleHeight = 3630
    ScaleWidth = 5865
    ShowInTaskbar = 0 'False
    StartUpPosition = 1 'CenterOwner
    Tag = "1077"
    Begin VB.PictureBox picIcon
        AutoSize = -1 'True
        BackColor = &H00C0C0C0&
        ClipControls = 0 'False
        Height = 540
        Left = 240
        Picture = "frmAbout.frx":0000
        ScaleHeight = 480
        ScaleMode = 0 'User
        ScaleWidth = 480
        TabIndex = 2
        TabStop = 0 'False
        Top = 240
        Width = 540
    End
    Begin VB.CommandButton cmdOK
        Cancel = -1 'True

```

```

Caption      = "OK"
Default      = -1 'True
Height       = 345
Left         = 4245
5 TabIndex    = 0
Tag          = "1079"
Top          = 2625
Width        = 1467
End
10 Begin VB.CommandButton cmdSysInfo
Caption      = "&System Info..."
Height       = 345
Left         = 4260
TabIndex     = 1
15 Tag        = "1078"
Top          = 3075
Width        = 1452
End
20 Begin VB.Label lblDescription
Caption      = "Providing at the pump advertising to convenience stores."
ForeColor    = &H00000000&
Height       = 1170
Left         = 1050
TabIndex     = 6
25 Tag        = "1083"
Top          = 1125
Width        = 4092
End
30 Begin VB.Label lblTitle
Caption      = "MediaVision"
ForeColor    = &H00000000&
Height       = 480
Left         = 1050
TabIndex     = 5
35 Tag        = "1082"
Top          = 240
Width        = 4092
End
40 Begin VB.Line Line1
BorderColor  = &H00808080&
BorderStyle = 6 'Inside Solid
Index        = 1
X1           = 225
X2           = 5657
45 Y1         = 2430
Y2           = 2430
End
50 Begin VB.Line Line1
BorderColor  = &H00FFFFFF&
BorderWidth  = 2
Index        = 0
X1           = 240
X2           = 5657
55 Y1         = 2445
Y2           = 2445

```

```

End
Begin VB.Label lblVersion
    Caption      = "Version 1.1"
    Height       = 225
5    Left        = 1050
    TabIndex     = 4
    Tag          = "1081"
    Top          = 780
    Width        = 4092
10    End
Begin VB.Label lblDisclaimer
    Caption      = "Warning: ... Patent or Copyright Warning here"
    ForeColor    = &H00000000&
    Height       = 825
15    Left       = 255
    TabIndex     = 3
    Tag          = "1080"
    Top          = 2625
    Width        = 3870
20    End
End

Public StartingAddress As String
Dim mbDontNavigateNow As Boolean
25 Private Sub Form_Load( )
    On Error Resume Next
    LoadResStrings Me
    Me.Show
    tbToolBar.Refresh
30    Form_Resize

    cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15

    If Len(StartingAddress) > 0 Then
35        cboAddress.Text = StartingAddress
        cboAddress.AddItem cboAddress.Text
        'try to navigate to the starting address
        timTimer.Enabled = True
        brwWebBrowser.Navigate StartingAddress
40    End If

End Sub

Private Sub brwWebBrowser_DownloadComplete( )
45    On Error Resume Next
    Me.Caption = brwWebBrowser.LocationName
End Sub

Private Sub brwWebBrowser_NavigateComplete2(ByVal pDisp As Object, URL As Variant)
50    On Error Resume Next
    Dim i As Integer
    Dim bFound As Boolean
    Me.Caption = brwWebBrowser.LocationName
    For i = 0 To cboAddress.ListCount - 1
55        If cboAddress.List (i) = brwWebBrowser.LocationURL Then

```

bFound = True
Exit For

End If

Next i

mbDontNavigateNow = True

If bFound Then

cboAddress.RemoveItem i

End If

cboAddress.AddItem brwWebBrowser.LocationURL, 0

cboAddress.ListIndex = 0

mbDontNavigateNow = False

End Sub

Private Sub cboAddress_Click ()

If mbDontNavigateNow Then Exit Sub

timTimer.Enabled = True

brwWebBrowser.Navigate cboAddress.Text

End Sub

Private Sub cboAddress_KeyPress(KeyAscii As Integer)

On Error Resume Next

If KeyAscii = vbKeyReturn Then

cboAddress_Click

End If

End Sub

Private Sub Form_Resize ()

On Error Resume Next

cboAddress.Width = Me.ScaleWidth - 100

brwWebBrowser.Width = Me.ScaleWidth - 100

brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top + picAddress.Height) - 100

End Sub

Private Sub timTimer_Timer ()

If brwWebBrowser.Busy = False Then

timTimer.Enabled = False

Me.Caption = brwWebBrowser.LocationName

Else

Me.Caption = "Working..."

End If

End Sub

Private Sub tbToolBar_ButtonClick(ByVal Button As Button)

On Error Resume Next

timTimer.Enabled = True

Select Case Button.Key

Case "Back"

brwWebBrowser.GoBack

Case "Forward"

brwWebBrowser.GoForward

Case "Refresh"


```

        brwWebBrowser.Refresh
    Case "Home"
        brwWebBrowser.GoHome
    Case "Search"
5       brwWebBrowser.GoSearch
    Case "Stop"
        timTimer.Enabled = False
        brwWebBrowser.Stop
        Me.Caption = brwWebBrowser.LocationName
10      End Select

    End Sub

VERSION 5.00
15      Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"
    Object = "{EAB22AC0-30C1-11CF-A7EB-0000C05BAE0B}#1.1#0"; "SHDOCVW.DLL"
    Begin VB.Form frmBrowser
        Caption           =       "MediaVision Web Browser"
        ClientHeight       =       5130
20         ClientLeft      =       3060
        ClientTop          =       3345
        ClientWidth        =       6540
        LinkTopic          =       "Form1"
        MDIChild           =       -1        'True
        ScaleHeight        =       5130
        ScaleWidth         =       6540
        ShowInTaskbar      =       0        'False
        Begin VB.Timer timTimer
            Enabled         =       0        'False
30             Interval    =       5
            Left           =       6180
            Top            =       1500
        End
        End
        Begin VB.PictureBox picAddress
35         Align           =       1        'Align Top
        BorderStyle         =       0        'None
        Height              =       675
        Left                =       0
        ScaleHeight         =       675
40         ScaleWidth      =       6540
        TabIndex            =       4
        TabStop             =       0        'False
        Top                 =       540
        Width               =       6540
        Begin VB.ComboBox cboAddress
45         Height          =       315
        Left               =       45
        TabIndex           =       2
        Text                =       "Combol"
50         Top             =       300
        Width              =       3795
        End
        Begin VB.Label lblAddress
55         Caption         =       "&Address:"
        Height              =       255

```

```

    Left = 45
    TabIndex = 1
    Tag = "1090"
    Top = 60
5    Width = 3075

    End
End
Begin MSComctlLib.Toolbar tbToolBar
    Align = 1 'Align Top
10    Height = 540
    Left = 0
    TabIndex = 3
    Top = 0
    Width = 6540
15    _ExtentX = 11536
    _ExtentY = 953
    ButtonWidth = 820
    ButtonHeight = 794
    Appearance = 1
20    ImageList = "imlToolBarIcons"
    _Version = 393216
    BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
        NumButtons = 6
        BeginProperty Button1 {66833FEA-8583-11D1-B16A-00C0F0283628}
25            Key = "Back"
            Object.ToolTipText = "1084"
            ImageIndex = 1
        EndProperty
        BeginProperty Button2 {66833FEA-8583-11D1-B16A-00C0F0283628}
30            Key = "Forward"
            Object.ToolTipText = "1085"
            ImageIndex = 2
        EndProperty
        BeginProperty Button3 {66833FEA-8583-11D1-B16A-00C0F0283628}
35            Key = "Stop"
            Object.ToolTipText = "1086"
            ImageIndex = 3
        EndProperty
        BeginProperty Button4 {66833FEA-8583-11D1-B16A-00C0F0283628}
40            Key = "Refresh"
            Object.ToolTipText = "1087"
            ImageIndex = 4
        EndProperty
        BeginProperty Button5 {66833FEA-8583-11D1-B16A-00C0F0283628}
45            Key = "Home"
            Object.ToolTipText = "1088"
            ImageIndex = 5
        EndProperty
        BeginProperty Button 6 {66833FEA-8583-11D1-B16A-00C0F0283628}
50            Key = "Search"
            Object.ToolTipText = "1089"
            ImageIndex = 6
        EndProperty
    EndProperty
EndProperty
55 End

```

```

Begin MSComctlLib.ImageList imlToolbarIcons
    Left      = 2670
    Top       = 2325
    _ExtentX  = 1005
5    _ExtentY  = 1005
    BackColor = -2147483643
    ImageWidth = 24
    ImageHeight = 24
    MaskColor  = 12632256
10    Version  = 393216
    BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}

        NumListImages = 6
    BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
15        Picture = {Binary}
        Key       = " "
    EndProperty
    BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
20        Picture = (Binary)
        Key       = " "
    EndProperty
    BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture = {Binary}
        Key       = " "
25    EndProperty
    BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture = {Binary}
        Key       = " "
    EndProperty
30    BeginProperty ListImage5 {2C247F27-8591-11D1-B16A-00C0F0283628}
        Picture = {Binary}
        Key       = " "
    EndProperty
    BeginProperty ListImage6 {12C247F27-8591-11D1-B16A-00C0F0283628}
35        Picture = {Binary}
        Key       = " "
    EndProperty
    EndProperty
End
40 Begin SHDocVwCtl.WebBrowser brwWebBrowser
    Height = 3734
    Left   = 50
    TabIndex = 0
    Top    = 1215
45    Width = 5393
    ExtentX = 9513
    ExtentY = 6586
    ViewMode = 0
    Offline = 0
50    Silent = 0
    RegisterAsBrowser = 0
    RegisterAsDropTarget = 1
    AutoArrange = 0 'False
    NoClientEdge = 0 'False
55    AlignLeft = 0 'False

```

```
ViewID = "(0057D0E0-3573-11CF-AE69-08002B2E1262)"
Location = ""
```

```
End
```

```
End
```

5

```
Private Sub rtfText_SelChange()
```

```
    fMainForm.tbToolBar.Buttons("Bold").Value = If(rtfText.SelBold, tbrPressed, tbrUnpressed)
```

```
    fMainForm.tbToolBar.Buttons("Italic").Value = If(rtfText.SelItalic, tbrPressed, tbrUnpressed)
```

```
    fMainForm.tbToolBar.Buttons("Underline").Value = If(rtfText.SelUnderline, tbrPressed, tbrUnpressed)
```

10

```
    fMainForm.tbToolBar.Buttons("Align Left").Value = If(rtfText.SelAlignment = rtfLeft, tbrPressed, tbrUnpressed)
```

```
    fMainForm.tbToolBar.Buttons("Center").Value = If(rtfText.SelAlignment = rtfCenter, tbrPressed, tbrUnpressed)
```

```
    fMainForm.tbToolBar.Buttons("Align Right").Value = If(rtfText.SelAlignment = rtfRight, tbrPressed, tbrUnpressed)
```

```
End Sub
```

15

```
Private Sub Form_Load()
```

```
    Form_Resize
```

```
End Sub
```

20

```
Private Sub Form_Resize()
```

```
    On Error Resume Next
```

```
    rtfText.Move 100, 100, Me.ScaleWidth - 200, Me.ScaleHeight - 200
```

```
    rtfText.RightMargin = rtfText.Width - 400
```

```
End Sub
```

25

```
VERSION 5.00
```

```
Object = "{3B7C8863-D78F-101B-B9B5-04021C009402}#1.2#0"; "RICHTX32.OCX"
```

```
Begin VB.Form frmDocument
```

```
    Caption = "MediaVision Document Editing"
```

```
    ClientHeight = 3195
```

30

```
    ClientLeft = 60
```

```
    ClientTop = 345
```

```
    ClientWidth = 4680
```

```
    LinkTopic = "Form1"
```

```
    MDIChild = -1 'True
```

35

```
    ScaleHeight = 3195
```

```
    ScaleWidth = 4680
```

```
    Begin RichTextLib.RichTextBox rtfText
```

```
        Height = 2000
```

```
        Left = 100
```

40

```
        TabIndex = 0
```

```
        Top = 100
```

```
        Width = 3000
```

```
        _ExtentX = 5292
```

```
        _ExtentY = 3519
```

45

```
        _Version = 393217
```

```
        Enabled = -1 'True
```

```
        ScrollBars = 3
```

```
        TextRTF = $"frmDocument.frx":0000
```

```
    End
```

50

```
End
```

```
Private Declare Function GetUserName Lib "advapi32.dll" Alias "GetUserNameA" (ByVal lpbuffer As String, nSize As Long) As Long
```

55

```
Public OK As Boolean
```

```

Private Sub Form_Load()
    Dim sBuffer As String
    Dim lSize As Long

5      LoadResStrings Me

    sBuffer = Space$(255)
    lSize = Len(sBuffer)
    Call GetUserName(sBuffer, lSize)
10     If lSize > 0 Then
        txtUserName.Text = Left$(sBuffer, lSize)
    Else
        txtUserName.Text = vbNullString
    End If
15     End Sub

Private Sub cmdCancel_Click()
    OK = False
    Me.Hide
20     End Sub

Private Sub cmdOK_Click()
    'ToDo: create test for correct password
    'check for correct password
25     If txtPassword.Text = "" Then
        OK = True
        Me.Hide
    Else
        MsgBox "Invalid Password, try again!", , "Login"
30     txtPassword.SetFocus
        txtPassword.SelStart = 0
        txtPassword.SelLength = Len(txtPassword.Text)
    End If
35     End Sub

VERSION 5.00
Begin VB.Form frmLogin
    BorderStyle   = 3 'Fixed Dialog
    Caption       = "MediaVision Login"
40    ClientHeight = 1590
    ClientLeft    = 45
    ClientTop     = 330
    ClientWidth   = 3750
    LinkTopic     = "Form1"
45    MaxButton    = 0 'False
    MinButton     = 0 'False
    ScaleHeight   = 1590
    ScaleWidth    = 3750
    ShowInTaskbar = 0 'False
50    StartUpPosition = 2 'CenterScreen
    Tag          = "1064"
    Begin VB.CommandButton cmdCancel
        Cancel     = -1 'True
        Caption     = "Cancel"
55        Height     = 360

```

```

Left      = 2100
TabIndex = 5
Tag       = "1068"
Top       = 1020
5 Width   = 1140
End
Begin VB.CommandButton cmdOK
Caption   = "OK"
Default  = -1 'True
10 Height = 360
Left     = 495
TabIndex = 4
Tag      = "1067"
Top      = 1020
15 Width  = 1140
End
Begin VB.TextBox txtPassword
Height   = 285
IMEMode  = 3 'DISABLE
20 Left   = 1305
PasswordChar = "*"
TabIndex = 1
Top      = 525
Width    = 2325
25 End
Begin VB.TextBox txtUserName
Height   = 285
Left     = 1305
30 TabIndex = 3
Top      = 135
Width    = 2325
End
35 Begin VB.Label lblLabels
Caption   = "&Password:"
Height   = 248
Index    = 1
Left     = 105
40 TabIndex = 0
Tag      = "1066"
Top      = 540
Width    = 1080
End
Begin VB.Label lblLabels
Caption   = "&User Name:"
45 Height   = 248
Index    = 0
Left     = 105
50 TabIndex = 2
Tag      = "1065"
Top      = 150
Width    = 1080
End
End
55 Private Declare Function SendMessage Lib "user32" Alias "SendMessageA" (ByVal hwnd As Long, ByVal wParam As

```

```

Long, ByVal wParam As Long, ByVal lParam As Any) As Long
Const EM_UNDO = &HC7
Private Declare Function OSWinHelp% Lib "user32" Alias "WinHelpA" (ByVal hwnd&, ByVal HelpFile$, ByVal
wCommand%, dwData As Any)

```

```

5 Private Sub MDIForm_Load()
    LoadResStrings Me
    Me.Left = GetSetting(App.Title, "Settings", "MainLeft", 1000)
    Me.Top = GetSetting(App.Title, "Settings", "MainTop", 1000)
10 Me.Width = GetSetting(App.Title, "Settings", "MainWidth", 6500)
    Me.Height = GetSetting(App.Title, "Settings", "MainHeight", 6500)
    LoadNewDoc
End Sub

```

```

15 Private Sub LoadNewDoc()
    Static IDocumentCount As Long
    Dim frmD As frmDocument
    IDocumentCount = IDocumentCount + 1
20 Set frmD = New frmDocument
    frmD.Caption = "Document " & IDocumentCount
    frmD.Show
End Sub

```

```

Private Sub SetVideo()
25 'Video files to be set up and sent via separate subroutine
    'these files will be sent via internet and satellite
End Sub

```

```

Private Sub Channel8()
30 'open channel 8 website and input weather forecast and seven
    'day planner graphic. These items will be combined with the
    'data stream and/or video stream and will be sent via subroutine
    'via internet and satellite.
End Sub

```

```

35 Private Sub MDIForm_Unload(Cancel As Integer)
    If Me.WindowState <> vbMinimized Then
        SaveSetting App.Title, "Settings", "MainLeft", Me.Left
        SaveSetting App.Title, "Settings", "MainTop", Me.Top
        SaveSetting App.Title, "Settings", "MainWidth", Me.Width
40 SaveSetting App.Title, "Settings", "MainHeight", Me.Height
    End If
End Sub

```

```

Private Sub tbToolBar_ButtonClick(ByVal Button As MSComCtlLib.Button)
45 On Error Resume Next
    Select Case Button.Key
        Case "New"
            LoadNewDoc
        Case "Open"
50 mnuFileOpen_Click
        Case "Save"
            mnuFileSave_Click
        Case "Print"
            mnuFilePrint_Click
55 Case "Cut"

```

```

        mnuEditCut_Click
    Case "Copy"
        mnuEditCopy_Click
    Case "Paste"
        mnuEditPaste_Click
    Case "Bold"
        ActiveForm.rtfText.SelBold = Not ActiveForm.rtfText.SelBold
        Button.Value = IIf(ActiveForm.rtfText.SelBold, tbrPressed, tbrUnpressed)
    Case "Italic"
        ActiveForm.rtfText.SelItalic = Not ActiveForm.rtfText.SelItalic
        Button.Value = IIf(ActiveForm.rtfText.SelItalic, tbrPressed, tbrUnpressed)
    Case "Underline"
        ActiveForm.rtfText.SelUnderline = Not ActiveForm.rtfText.SelUnderline
        Button.Value = IIf(ActiveForm.rtfText.SelUnderline, tbrPressed, tbrUnpressed)
    Case "Align Left"
        ActiveForm.rtfText.SelAlignment = rtfLeft
    Case "Center"
        ActiveForm.rtfText.SelAlignment = rtfCenter
    Case "Align Right"
        ActiveForm.rtfText.SelAlignment = rtfRight
    Case "Camera"
        'ToDo: Add 'Camera' button code.
        MsgBox "Add 'Camera' button code."
    Case "Delete"
        'ToDo: Add 'Delete' button code.
        MsgBox "Add 'Delete' button code."
    Case "Find"
        'ToDo: Add 'Find' button code.
        MsgBox "Add 'Find' button code."
    Case "View Details"
        'ToDo: Add 'View Details' button code.
        MsgBox "Add 'View Details' button code."
    Case "Help What's This"
        'ToDo: Add 'Help What's This' button code.
        MsgBox "Add 'Help What's This' button code."
    Case "Help"
        'ToDo: Add 'Help' button code.
        MsgBox "Add 'Help' button code."
End Select
End Sub

Private Sub mnuHelpAbout_Click()
    frmAbout.Show vbModal, Me
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
    Dim nRet As Integer

    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project.", vbInformation,
        Me.Caption
    Else

```



```

On Error Resume Next
nRet = OSWinHelp(Me.hwnd, App.HelpFile, 261, 0)
If Err Then
    MsgBox Err.Description
5    End If
End If

End Sub

10 Private Sub mhuHelpContents_Click()
    Dim nRet as Integer

    'if there is no helpfile for this project display a message to the user
    'you can set the HelpFile for your application in the
15    'Project Properties dialog
    If Len(App.HelpFile) = 0 Then
        MsgBox "Unable to display Help Contents. There is no Help associated with this project."
        , vbInformation, Me.Caption
    Else
20        On error Resume Next
        nRet = OSWinHelp(Me.hwnd, App.HelpFile, 3, 0)
        If Err Then
            MsgBox Err.Description
        End If
25    End If

End Sub

Private Sub mnuWindowArrangeIcons_Click()
30    Me.Arrange vbArrangeIcons
End Sub

Private Sub mnuWindowTileVertical_Click()
    Me.Arrange vbTileVertical
35 End Sub

Private Sub mnuWindowTileHorizontal_Click()
    Me.Arrange vbTileHorizontal
End Sub

40 Private Sub mnuWindowCascade_Click()
    Me.Arrange vbCascade
End Sub

45 Private Sub mnuWindowNewWindow_Click()
    LoadNewDoc
End Sub

Private Sub LoadInternet()
    open"http://www.msn.com/user8982734/23486219"
50    'input news, stocks etc. through subroutine
    'output information to internet file through separate subroutine
    'output information to satellite file through seaprte subroutine

End Sub
55

```

```

Private Sub mnuToolsOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

5 Private Sub mnuViewWebBrowser_Click()
    Dim frmB As New frmBrowser
    frmB.StartingAddress = "http://www.mediavision.com"
    frmB.Show
End Sub

10 Private Sub mnuViewOptions_Click()
    frmOptions.Show vbModal, Me
End Sub

15 Private Sub mnuViewRefresh_Click()
    'ToDo: Add 'mnuViewRefresh_Click' code.
    MsgBox "Add 'mnuViewRefresh_Click' code."
End Sub

20 Private Sub mnuViewStatusBar_Click()
    mnuViewStatusBar.Checked = Not mnuViewStatusBar.Checked
    sbStatusBar.Visible = mnuViewStatusBar.Checked
End Sub

25 Private Sub mnuViewToolbar_Click()
    mnuViewToolbar.Checked = Not mnuViewToolbar.Checked
    tbToolBar.Visible = mnuViewToolbar.Checked
End Sub

30 Private Sub mnuEditPasteSpecial_Click()
    'ToDo: Add 'mnuEditPasteSpecial_Click' code.
    MsgBox "Add 'mnuEditPasteSpecial_Click' code."
End Sub

35 Private Sub mnuEditPaste_Click()
    On Error Resume Next
    ActiveForm.rtfText.SelRTF = Clipboard.GetText
End Sub

40 Private Sub mnuEditCopy_Click()
    On Error Resume Next
    Clipboard.SetText ActiveForm.rtfText.SelRTF
End Sub

45 Private Sub mnuEditCut_Click()
    On Error Resume Next
    Clipboard.SetText ActiveForm.rtfText.SelRTF
    ActiveForm.rtfText.SelText = vbNullString
End Sub

50 Private Sub mnuEditUndo_Click()
    'ToDo: Add 'mnuEditUndo_Click' code.

```

```
MsgBox "Add 'mnuEditUndo_Click' code."
End Sub
```

```
5 Private Sub mnuFileExit_Click()
    'unload the form
    Unload Me
```

```
End Sub
```

10

```
Private Sub mnuFileSend_Click()
    'ToDo: Add 'mnuFileSend_Click' code.
    MsgBox "Add 'mnuFileSend_Click' code."
End Sub
```

15

```
Private Sub mnuFilePrint_Click()
    On Error Resume Next
    If ActiveForm Is Nothing Then Exit Sub
```

20

```
With dlgCommonDialog
    .DialogTitle = "Print"
    .CancelError = True
    .Flags = cdIPDReturnDC + cdIPDNoPageNums
    If ActiveForm.rtfText.SelLength = 0 Then
    .Flags = .Flags + cdIPDAllPages
    Else
    .Flags = .Flags + cdIPDSelection
    End If
    .ShowPrinter
    If Err <> MSComDlg.cdlCancel Then
        ActiveForm.rtfText.SelPrint .hDC
    End If
End With
```

25
30
35
40
45
50
55

```
End Sub
```

```
Private Sub mnuFilePrintPreview_Click()
    'ToDo: Add 'mnuFilePrintPreview_Click' code.
    MsgBox "Add 'mnuFilePrintPreview_Click' code."
End Sub
```

40

```
Private Sub mnuFilePageSetup_Click()
    On Error Resume Next
    With dlgCommonDialog
    .DialogTitle = "Page Setup"
    .CancelError = True
    .ShowPrinter
    End With
```

45

```
50 End Sub
```

```
Private Sub mnuFileProperties_Click()
    'ToDo: Add 'mnuFileProperties_Click' code.
    MsgBox "Add 'mnuFileProperties_Click' code."
End Sub
```

55

```

Private Sub mnuFileSaveAll_Click()
    'ToDo: Add 'mnuFileSaveAll_Click' code.
    MsgBox "Add 'mnuFileSaveAll_Click' code."
End Sub

```

```

Private Sub mnuFileSaveAs_Click()
    Dim sFile As String

    If ActiveForm Is Nothing Then Exit Sub

```

```

    With dlgCommonDialog
        .DialogTitle = "Save As"
        .CancelError = False
        'ToDo: set the flags and attributes of the common dialog control
        .Filter = "All Files (*.*)|*.*"
        .ShowSave
        If Len(.FileName) = 0 Then
            Exit Sub
        End If
        sFile = .FileName
    End With
    ActiveForm.Caption = sFile
    ActiveForm.rtfText.SaveFile sFile

```

```

End Sub

```

```

Private Sub mnuFileSave_Click()
    Dim sFile As String
    If Left$(ActiveForm.Caption, 8) = "Document" Then
        With dlgCommonDialog
            .DialogTitle = "Save"
            .CancelError = False
            'ToDo: set the flags and attributes of the common dialog control
            .Filter = "All Files (*.*)|*.*"
            .ShowSave
            If Len(.FileName) = 0 Then
                Exit Sub
            End If
            sFile = .FileName
        End With
        ActiveForm.rtfText.SaveFile sFile
    Else
        sFile = ActiveForm.Caption
        ActiveForm.rtfText.SaveFile sFile
    End If

```

```

End Sub

```

```

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

```

```

Private Sub mnuFileOpen_Click()
    Dim sFile As String

```

If ActiveForm Is Nothing Then LoadNewDoc

With dlgCommonDialog

.DialogTitle = "Open"

.CancelError = False

'ToDo: set the flags and attributes of the common dialog control

.Filter = "All Files (*.*)|*.*"

.ShowOpen

If Len(.FileName) = 0 Then

Exit Sub

End If

sFile = .FileName

End With

ActiveForm.rtfText.LoadFile sFile

ActiveForm.Caption = sFile

End Sub

Private Sub mnuFileNew_Click()

LoadNewDoc

End Sub

VERSION 5.00

Object = "{F9043C88-F6F2-101A-A3C9-08002B2F49FB}#1.2#0"; "COMDLG32.OCX"

Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0"; "MSCOMCTL.OCX"

Begin VB.MDIForm frmMain

Caption = "MediaVision"

ClientHeight = 3195

ClientLeft = 60

ClientTop = 630

ClientWidth = 4680

LinkTopic = "MDIForm1"

StartupPosition = 3 'Windows Default

Begin MSComctlLib.StatusBar sbStatusBar

Align = 2 'Align Bottom

Height = 270

Left = 0

TabIndex = 0

Top = 10560

Width = 15240

_ExtentX = 26882

_ExtentY = 476

_Version = 393216

BeginProperty Panels {8E3867A5-8586-11D1-B16A-00C0F0283628}

NumPanels = 3

BeginProperty Panel1 {8E3867AB-8586-11D1-B16A-00C0F0283628}

AutoSize = 1

Object.Width = 21246

Text = "Status"

TextSave = "Status"

EndProperty

BeginProperty Panel2 {8E3867AB-8586-11D1-B16A-00C0F0283628}

Style = 6

AutoSize = 2

TextSave = "4/17/00"

```

EndProperty
BeginProperty Panel3 {8E3867AB-8586-11D1-B16A-00C0F0283628}
  Style      = 5
  AutoSize   = 2
  TextSave   = "11:14 AM"
EndProperty
EndProperty
End
Begin MSComDlg.CommonDialog dlgCommonDialog
  Left       = 720
  Top        = 6240
  _ExtentX   = 847
  _ExtentY   = 847
  _Version   = 393216
End
Begin MSComctlLib.ImageList imlToolbarIcons
  Left       = 600
  Top        = 6960
  _ExtentX   = 1005
  _ExtentY   = 1005
  BackColor  = -2147483643
  ImageWidth  = 16
  ImageHeight = 16
  MaskColor   = 12632256
  _Version   = 393216
BeginProperty Images {2C247F25-8591-11D1-B16A-00C0F0283628}
  NumListImages = 19
  BeginProperty ListImage1 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":0000
    Key         = "New"
  EndProperty
  BeginProperty ListImage2 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":0112
    Key         = "Open"
  EndProperty
  BeginProperty ListImage3 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":0224
    Key         = "Save"
  EndProperty
  BeginProperty ListImage4 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":0336
    Key         = "Print"
  EndProperty
  BeginProperty ListImage5 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":0448
    Key         = "Cut"
  EndProperty
  BeginProperty ListImage6 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":055A
    Key         = "Copy"
  EndProperty
  BeginProperty ListImage7 {2C247F27-8591-11D1-B16A-00C0F0283628}
    Picture     = "frmMain.frx":066C
    Key         = "Paste"
  EndProperty
EndProperty

```

```

BeginProperty ListImage8 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":077E
  Key          = "Bold"
EndProperty
5 BeginProperty ListImage9 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":0890
  Key          = "Italic"
EndProperty
10 BeginProperty ListImage10 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":09A2
  Key          = "Underline"
EndProperty
15 BeginProperty ListImage11 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":0AB4
  Key          = "Align Left"
EndProperty
20 BeginProperty ListImage12 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":0BC6
  Key          = "Center"
EndProperty
25 BeginProperty ListImage13 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":0CD8
  Key          = "Align Right"
EndProperty
30 BeginProperty ListImage14 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":0DEA
  Key          = "Camera"
EndProperty
35 BeginProperty ListImage15 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":0EFC
  Key          = "Delete"
EndProperty
40 BeginProperty ListImage16 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":100E
  Key          = "Find"
EndProperty
45 BeginProperty ListImage17 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":1120
  Key          = "View Details"
EndProperty
50 BeginProperty ListImage18 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":1232
  Key          = "Help What's This"
EndProperty
55 BeginProperty ListImage19 {2C247F27-8591-11D1-B16A-00C0F0283628}
  Picture      = "frmMain.frx":1344
  Key          = "Help"
EndProperty
EndProperty
End
Begin MSComctlLib.Toolbar tbToolBar
  Align        = 1 'Align Top
  Height       = 420
  Left         = 0
  TabIndex     = 1

```

```

Top      = 0
Width    = 15240
_ExtentX = 26882
_ExtentY = 741
5 ButtonWidth = 609
Appearance = 1
ImageList = "imlToolbarIcons"
_Version = 393216
BeginProperty Buttons {66833FE8-8583-11D1-B16A-00C0F0283628}
10 NumButtons = 23
   BeginProperty Button1 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Key = "New"
       Object.ToolTipText = "1037"
       ImageKey = "New"
15 EndProperty
   BeginProperty Button2 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Key = "Open"
       Object.ToolTipText = "1038"
       ImageKey = "Open"
20 EndProperty
   BeginProperty Button3 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Key = "Save"
       Object.ToolTipText = "1039"
       ImageKey = "Save"
25 EndProperty
   BeginProperty Button4 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Style = 3
   EndProperty
   BeginProperty Button5 {66833FEA-8583-11D1-B16A-00C0F0283628}
30 Key = "Print"
       Object.ToolTipText = "1040"
       ImageKey = "Print"
   EndProperty
   BeginProperty Button6 {66833FEA-8583-11D1-B16A-00C0F0283628}
35 Style = 3
   EndProperty
   BeginProperty Button7 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Key = "Cut"
       Object.ToolTipText = "1041"
       ImageKey = "Cut"
40 EndProperty
   BeginProperty Button8 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Key = "Copy"
       Object.ToolTipText = "1042"
       ImageKey = "Copy"
45 EndProperty
   BeginProperty Button9 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Key = "Paste"
       Object.ToolTipText = "1043"
       ImageKey = "Paste"
50 EndProperty
   BeginProperty Button10 {66833FEA-8583-11D1-B16A-00C0F0283628}
       Style = 3
   EndProperty
55 BeginProperty Button11 {66833FEA-8583-11D1-B16A-00C0F0283628}

```



```

Key      = "Bold"
Object.ToolTipText = "1044"
ImageKey  = "Bold"
EndProperty
5 BeginProperty Button12 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Italic"
  Object.ToolTipText = "1045"
  ImageKey  = "Italic"
EndProperty
10 BeginProperty Button13 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Underline"
  Object.ToolTipText = "1046"
  ImageKey  = "Underline"
EndProperty
15 BeginProperty Button14 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Style     = 3
EndProperty
BeginProperty Button15 {66833FEA-8583-11D1-B16A-00C0F0283628}
20   Key      = "Align Left"
  Object.ToolTipText = "1047"
  ImageKey  = "Align Left"
  Style     = 2
EndProperty
BeginProperty Button16 {66833FEA-8583-11D1-B16A-00C0F0283628}
25   Key      = "Center"
  Object.ToolTipText = "1048"
  ImageKey  = "Center"
  Style     = 2
EndProperty
30 BeginProperty Button17 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Align Right"
  Object.ToolTipText = "1049"
  ImageKey  = "Align Right"
  Style     = 2
35 EndProperty
BeginProperty Button18 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Camera"
  Object.ToolTipText = "1050"
  ImageKey  = "Camera"
40 EndProperty
BeginProperty Button19 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Delete"
  Object.ToolTipText = "1051"
  ImageKey  = "Delete"
45 EndProperty
BeginProperty Button20 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Find"
  Object.ToolTipText = "1052"
  ImageKey  = "Find"
50 EndProperty
BeginProperty Button21 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "View Details"
  Object.ToolTipText = "1053"
  ImageKey  = "View Details"
55 Style     = 2

```

```

EndProperty
BeginProperty Button22 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Help What's This"
  Object.ToolTipText = "1054"
5   ImageKey = "Help What's This"
EndProperty
BeginProperty Button23 {66833FEA-8583-11D1-B16A-00C0F0283628}
  Key      = "Help"
  Object.ToolTipText = "1055"
10  ImageKey = "Help"
EndProperty
EndProperty
End
Begin VB.Menu mnuFile
15  Caption      = "1000"
Begin VB.Menu mnuFileNew
  Caption      = "1001"
  Shortcut     = ^N
End
20  Begin VB.Menu mnuFileOpen
    Caption      = "1002"
    Shortcut     = ^O
End
Begin VB.Menu mnuFileClose
25  Caption      = "1003"
End
Begin VB.Menu mnuFileBar0
  Caption      = "-"
End
30  Begin VB.Menu mnuFileSave
    Caption      = "1004"
End
Begin VB.Menu mnuFileSaveAs
  Caption      = "1005"
35  End
Begin VB.Menu mnuFileSaveAll
  Caption      = "1006"
End
40  Begin VB.Menu mnuFileBar1
    Caption      = "-"
End
Begin VB.Menu mnuFileProperties
  Caption      = "1007"
End
45  Begin VB.Menu mnuFileBar2
    Caption      = "-"
End
Begin VB.Menu mnuFilePageSetup
  Caption      = "1008"
50  End
Begin VB.Menu mnuFilePrintPreview
  Caption      = "1009"
End
55  Begin VB.Menu mnuFilePrint
    Caption      = "1010"

```

```

End
Begin VB.Menu mnuFileBar3
    Caption    = "-"
End
5  Begin VB.Menu mnuFileSend
    Caption    = "1011"
End
    Begin VB.Menu mnuFileBar4
        Caption    = "-"
10  End
    Begin VB.Menu mnuFileMRU
        Caption    = ""
        Index      = 1
        Visible    = 0 'False
15  End
    Begin VB.Menu mnuFileMRU
        Caption    = ""
        Index      = 2
        Visible    = 0 'False
20  End
    Begin VB.Menu mnuFileMRU
        Caption    = ""
        Index      = 3
        Visible    = 0 'False
25  End
    Begin VB.Menu mnuFileBar5
        Caption    = "-"
        Visible    = 0 'False
30  End
    Begin VB.Menu mnuFileExit
        Caption    = "1012"
    End
End
35  Begin VB.Menu mnuEdit
    Caption    = "1013"
    Begin VB.Menu mnuEditUndo
        Caption    = "1014"
    End
    Begin VB.Menu mnuEditBar0
40  Caption    = "-"
    End
    Begin VB.Menu mnuEditCut
        Caption    = "1015"
        Shortcut    = ^X
45  End
    Begin VB.Menu mnuEditCopy
        Caption    = "1016"
        Shortcut    = ^C
    End
50  Begin VB.Menu mnuEditPaste
        Caption    = "1017"
        Shortcut    = ^V
    End
    Begin VB.Menu mnuEditPasteSpecial
55  Caption    = "1018"

```

```

End
End
Begin VB.Menu mnuView
Caption      = "1019"
Begin VB.Menu mnuViewToolBar
Caption      = "1020"
Checked      = -1 'True
End
Begin VB.Menu mnuViewStatusBar
Caption      = "1021"
Checked      = -1 'True
End
Begin VB.Menu mnuViewBar0
Caption      = "-"
End
Begin VB.Menu mnuViewRefresh
Caption      = "1022"
End
Begin VB.Menu mnuViewOptions
Caption      = "1023"
End
Begin VB.Menu mnuViewWebBrowser
Caption      = "1024"
End
End
Begin VB.Menu mnuTools
Caption      = "1025"
Begin VB.Menu mnuToolsOptions
Caption      = "1026"
End
End
Begin VB.Menu mnuWindow
Caption      = "1027"
WindowList   = -1 'True
Begin VB.Menu mnuWindowNewWindow
Caption      = "1028"
End
Begin VB.Menu mnuWindowBar0
Caption      = "-"
End
Begin VB.Menu mnuWindowCascade
Caption      = "1029"
End
Begin VB.Menu mnuWindowTileHorizontal
Caption      = "1030"
End
Begin VB.Menu mnuWindowTileVertical
Caption      = "1031"
End
Begin VB.Menu mnuWindowArrangeIcons
Caption      = "1032"
End
End
Begin VB.Menu mnuHelp
Caption      = "1033"

```

```

Begin VB.Menu mnuHelpContents
    Caption      = "1034"
End
Begin VB.Menu mnuHelpSearchForHelpOn
5    Caption      = "1035"
End
Begin VB.Menu mnuHelpBar0
    Caption      = "-"
End
10    Begin VB.Menu mnuHelpAbout
        Caption      = "1036"
    End
End
End
15    Private Sub Form_Load()
        LoadResStrings Me
    End Sub

20    Private Sub cmdApply_Click()
        'ToDo: Add 'cmdApply_Click' code.
        MsgBox "Apply Code goes here to set options w/o closing dialog!"
    End Sub

25    Private Sub cmdCancel_Click()
        Unload Me
    End Sub

30    Private Sub cmdOK_Click()
        'ToDo: Add 'cmdOK_Click' code.
        MsgBox "Code goes here to set options and close dialog!"
        Unload Me
    End Sub

35    Private Sub Form_KeyDown(KeyCode As Integer, Shift As Integer)
        Dim i As Integer
        i = tbsOptions.SelectedItem.Index
        'handle ctrl+tab to move to the next tab
        If (Shift And 3) = 2 And KeyCode = vbKeyTab Then
40            If i = tbsOptions.Tabs.Count Then
                'last tab so we need to wrap to tab 1
                Set tbsOptions.SelectedItem = tbsOptions.Tabs(1)
            Else
                'increment the tab
45                Set tbsOptions.SelectedItem = tbsOptions.Tabs(i + 1)
            End If
        ElseIf (Shift And 3) = 3 And KeyCode = vbKeyTab Then
            If i = 1 Then
50                'last tab so we need to wrap to tab 1
                Set tbsOptions.SelectedItem = tbsOptions.Tabs(tbsOptions.Tabs.Count)
            Else
                'increment the tab
                Set tbsOptions.SelectedItem = tbsOptions.Tabs(i - 1)
            End If
55        End If
    End Sub

```

End Sub

Private Sub tbsOptions_Click()

```
5      Dim i As Integer
      'show and enable the selected tab's controls
      'and hide and disable all others
      For i = 0 To tbsOptions.Tabs.Count - 1
          If i = tbsOptions.SelectedItem.Index - 1 Then
10             picOptions(i).Left = 210
             picOptions(i).Enabled = True
          Else
             picOptions(i).Left = -20000
             picOptions(i).Enabled = False
15          End If
      Next
```

End Sub

```
20  VERSION 5.00
      Object = "{831FDD16-0C5C-11D2-A9FC-0000F8754DA1}#2.0#0";
      "MSCOMCTL.OCX"
      Begin VB.Form frmOptions
25         BorderStyle   = 3 'Fixed Dialog
         Caption       = "MediaVision Options"
         ClientHeight   = 5040
         ClientLeft     = 45
         ClientTop      = 330
         ClientWidth    = 6150
30         KeyPreview    = -1 'True
         LinkTopic      = "Form1"
         MaxButton      = 0 'False
         MinButton      = 0 'False
         ScaleHeight     = 5040
35         ScaleWidth    = 6150
         ShowInTaskbar  = 0 'False
         StartUpPosition = 1 'CenterOwner
         Tag            = "1069"
         Begin VB.CommandButton cmdOK
40             Caption   = "OK"
             Height     = 375
             Left       = 2490
             TabIndex   = 1
             Tag        = "1076"
45             Top       = 4455
             Width      = 1095
         End
         Begin VB.CommandButton cmdCancel
50             Cancel    = -1 'True
             Caption    = "Cancel"
             Height     = 375
             Left       = 3720
             TabIndex   = 3
             Tag        = "1075"
55             Top       = 4455
         End
```

```

Width      = 1095
End
Begin VB.CommandButton cmdApply
Caption     = "&Apply"
5   Height   = 375
    Left     = 4920
    TabIndex = 5
    Tag      = "1074"
    Top      = 4455
10  Width    = 1095
End
Begin VB.PictureBox picOptions
BorderStyle = 0 'None
15  Height   = 3780
    Index    = 3
    Left     = -20000
    ScaleHeight = 3840.968
    ScaleMode = 0 'User
    ScaleWidth  = 5745.64
20  TabIndex = 7
    TabStop  = 0 'False
    Top      = 480
    Width    = 5685
Begin VB.Frame fraSample4
25  Caption   = "Sample 4"
    Height    = 2022
    Left      = 505
    TabIndex  = 11
    Tag       = "1073"
30  Top       = 502
    Width     = 2033
End
End
Begin VB.PictureBox picOptions
35  BorderStyle = 0 'None
    Height      = 3780
    Index       = 2
    Left        = -20000
    ScaleHeight = 3840.968
40  ScaleMode  = 0 'User
    ScaleWidth  = 5745.64
    TabIndex    = 9
    TabStop     = 0 'False
    Top         = 480
45  Width      = 5685
Begin VB.Frame fraSample3
    Caption     = "Sample 3"
    Height      = 2022
    Left        = 406
50  TabIndex   = 10
    Tag         = "1072"
    Top         = 403
    Width       = 2033
End
55 End
End

```

```

Begin VB.PictureBox picOptions
  BorderStyle = 0 'None
  Height      = 3780
  Index       = 1
5   Left      = -20000
  ScaleHeight = 3840.968
  ScaleMode   = 0 'User
  ScaleWidth  = 5745.64
  TabIndex    = 6
10  TabStop   = 0 'False
  Top         = 480
  Width       = 5685
  Begin VB.Frame fraSample2
    Caption    = "Sample 2"
15  Height    = 2022
    Left      = 307
    TabIndex  = 8
    Tag       = "1071"
    Top       = 305
20  Width    = 2033
  End
End
Begin VB.PictureBox picOptions
  BorderStyle = 0 'None
25  Height    = 3780
  Index       = 0
  Left       = 210
  ScaleHeight = 3840.968
  ScaleMode   = 0 'User
30  ScaleWidth = 5745.64
  TabIndex    = 2
  TabStop     = 0 'False
  Top         = 480
  Width       = 5685
35  Begin VB.Frame fraSample1
    Caption    = "Sample 1"
    Height    = 2022
    Left      = 208
    TabIndex  = 4
40  Tag       = "1070"
    Top       = 207
    Width     = 2033
  End
End
45  Begin MSComctlLib.TabStrip tbsOptions
  Height      = 4245
  Left        = 120
  TabIndex    = 0
  Top         = 120
50  Width     = 5895
  _ExtentX    = 10398
  _ExtentY    = 7488
  _Version    = 393216
  BeginProperty Tabs {1EFB6598-857C-11D1-B16A-00C0F0283628}
55  NumTabs    = 4

```



```

BeginProperty Tab1 {1EFB659A-857C-11D1-B16A-00C0F0283628}
  Caption      = "Group 1"
  ImageVarType = 2
EndProperty
5  BeginProperty Tab2 {1EFB659A-857C-11D1-B16A-00C0F0283628}
  Caption      = "Group 2"
  ImageVarType = 2
EndProperty
10 BeginProperty Tab3 {1EFB659A-857C-11D1-B16A-00C0F0283628}
  Caption      = "Group 3"
  ImageVarType = 2
EndProperty
15 BeginProperty Tab4 {1EFB659A-857C-11D1-B16A-00C0F0283628}
  Caption      = "Group 4"
  ImageVarType = 2
EndProperty
EndProperty
End
End
20
Private Sub Form_Load()
  LoadResStrings Me
  lblVersion.Caption = "Version " & App.Major & "." & App.Minor & "." & App.Revision
  lblProductName.Caption = App.Title
25 End Sub

VERSION 5.00
Begin VB.Form frmSplash
  BorderStyle  = 3 'Fixed Dialog
30  ClientHeight = 4710
  ClientLeft   = 45
  ClientTop    = 45
  ClientWidth  = 7455
  ControlBox   = 0 'False
35  LinkTopic   = "Form1"
  MaxButton    = 0 'False
  MinButton    = 0 'False
  ScaleHeight  = 4710
  ScaleWidth   = 7455
40  ShowInTaskbar = 0 'False
  StartUpPosition = 2 'CenterScreen
  Visible      = 0 'False
  Begin VB.Frame fraMainFrame
45    Height      = 4590
    Left         = 45
    TabIndex     = 0
    Top          = -15
    Width        = 7380
    Begin VB.PictureBox picLogo
50      Height      = 2385
      Left         = 510
      Picture       = "frmSplash.frx":0000
      ScaleHeight  = 2325
      ScaleWidth   = 1755
55      TabIndex   = 1

```

```

Top      = 855
Width    = 1815
End
Begin VB.Label lblProductName
5  AutoSize    = -1 'True
   Caption     = "Product Name?"
   BeginProperty Font
       Name      = "MS Sans Serif"
       Size      = 29.25
10  Charset     = 0
       Weight    = 700
       Underline  = 0 'False
       Italic     = 0 'False
       Strikethrough = 0 'False
15  EndProperty
   Height      = 720
   Left        = 2670
   TabIndex    = 8
   Tag         = "1062"
20  Top        = 1200
   Width       = 4395
End
Begin VB.Label lblCompanyProduct
25  AutoSize    = -1 'True
   Caption     = "MediaVision"
   BeginProperty Font
       Name      = "MS Sans Serif"
       Size      = 18
30  Charset     = 0
       Weight    = 700
       Underline  = 0 'False
       Italic     = 0 'False
       Strikethrough = 0 'False
35  EndProperty
   Height      = 435
   Left        = 2505
   TabIndex    = 7
   Tag         = "1061"
40  Top        = 765
   Width       = 2175
End
Begin VB.Label lblPlatform
45  Alignment   = 1 'Right Justify
   AutoSize    = -1 'True
   Caption     = "Platform? Satellite or Internet"
   BeginProperty Font
       Name      = "MS Sans Serif"
       Size      = 13.5
50  Charset     = 0
       Weight    = 700
       Underline  = 0 'False
       Italic     = 0 'False
       Strikethrough = 0 'False
55  EndProperty
   Height      = 360

```

```

Left      = 3000
TabIndex  = 6
Tag       = "1060"
Top       = 2400
5 Width   = 4005
End
Begin VB.Label lblVersion
Alignment = 1 'Right Justify
AutoSize  = -1 'True
10 Caption = "Version 1.1"
BeginProperty Font
Name      = "MS Sans Serif"
Size      = 12
Charset   = 0
15 Weight  = 700
Underline = 0 'False
Italic    = 0 'False
Strikethrough = 0 'False
EndProperty
20 Height  = 300
Left      = 5625
TabIndex  = 5
Tag       = "1059"
Top       = 2760
25 Width  = 1380
End
Begin VB.Label lblWarning
Caption    = "Patent or Copyright Warning Here"
30 Height  = 195
Left      = 300
TabIndex  = 2
Tag       = "1058"
Top       = 3720
35 Width  = 6855
End
Begin VB.Label lblCompany
Caption    = "MediaVision, Inc."
40 Height  = 255
Left      = 4710
TabIndex  = 4
Tag       = "1057"
Top       = 3330
Width     = 2415
End
45 Begin VB.Label lblCopyright
Caption    = "Copyright 2000"
Height    = 255
Left      = 4710
50 TabIndex = 3
Tag       = "1056"
Top       = 3120
Width     = 2415
End
End
55 End

```

Public fMainForm As frmMain

Sub Main()

Dim fLogin As New frmLogin

fLogin.Show vbModal

If Not fLogin.OK Then

 'Login Failed so exit app

 End

End If

Unload fLogin

frmSplash.Show

frmSplash.Refresh

Set fMainForm = New frmMain

Load fMainForm

Unload frmSplash

fMainForm.Show

End Sub

Sub LoadResStrings(frm As Form)

On Error Resume Next

Dim ctl As Control

Dim obj As Object

Dim fnt As Object

Dim sCtlType As String

Dim nVal As Integer

'set the form's caption

frm.Caption = LoadResString(CInt(frm.Tag))

'set the font

Set fnt = frm.Font

fnt.Name = LoadResString(20)

fnt.Size = CInt(LoadResString(21))

'set the controls' captions using the caption

'property for menu items and the Tag property

'for all other controls

For Each ctl In frm.Controls

 Set ctl.Font = fnt

 sCtlType = TypeName(ctl)

 If sCtlType = "Label" Then

 ctl.Caption = LoadResString(CInt(ctl.Tag))

 ElseIf sCtlType = "Menu" Then

 ctl.Caption = LoadResString(CInt(ctl.Caption))

 ElseIf sCtlType = "TabStrip" Then

 For Each obj In ctl.Tabs

 obj.Caption = LoadResString(CInt(obj.Tag))

 obj.ToolTipText = LoadResString(CInt(obj.ToolTipText))

 Next

 ElseIf sCtlType = "Toolbar" Then

 For Each obj In ctl.Buttons

```

    obj.ToolTipText = LoadResString(CInt(obj.ToolTipText))
Next
ElseIf sCtlType = "ListView" Then
    For Each obj In ctl.ColumnHeaders
        obj.Text = LoadResString(CInt(obj.Tag))
    Next
Else
    nVal = 0
    nVal = Val(ctl.Tag)
    If nVal > 0 Then ctl.Caption = LoadResString(nVal)
    nVal = 0
    nVal = Val(ctl.ToolTipText)
    If nVal > 0 Then ctl.ToolTipText = LoadResString(nVal)
End If
Next
End Sub

```

Figures 2 and 3 are flow charts illustrating the software functionality of Phase I software ticker and image processing components (a.k.a. ticker and image components, respectively). In the invention's preferred embodiment, said Phase I software operates in a Windows® environment, however the invention's programming logic is neither platform specific nor dependent.

The data acquisition central processing unit can utilize a dial-up connection, but normally functions with a high-speed Internet connection and can communicate directly with the remote site central processing units, or can upload information via FTP, HTTP, or HTTPS protocols.

Figure 2 illustrates a software processing component which facilitates the retrieval, processing, and transmitting of "Ticker" information. Such information represents a variety of viewer-essential information including, but not limited to, current national news headlines, local news headlines, national and local sports headlines, national and local sports scores, stock quotes, stock market indices and news 2.05.

The Phase I processing component is capable of retrieving and inputting 2.07 information from the Internet, virtual private network (VPN), or direct modem connection and can be customized for an individual client. For example, if the client is a bank the ticker could retrieve current CD rates and other client-specific information.

The data acquisition central processing unit contains a database which segregates remote site central processing units by assigning each group and sub-group codes 2.04. Such codes can be assigned based upon geographic or other criteria 2.12 and this processing component allows for complete user-definable groups and sub-groups.

5 These codes are utilized to determine information to be sent to each remote computer. For example, each or every remote computer within a metropolitan area may be sent local news and weather information specific to that area 2.12.

Once ticker information for each remote computer has been generated 2.07, 2.08, it is transmitted either directly to the remote computer (via direct modem connection, VPN, or FTP) 2.15 or is transmitted to a private web server 2.05. A web server acts as a repository of information which is continually updated and available for the remote site central processing units to download.

10 Figure 3 illustrates the program processing means steps which facilitate the processing and transmitting of image files. The invention's image files consist of, but are not limited to, full-motion video (FMV), graphics, animations, and streaming video. Audio is also a component of these file types and can be integrated into the file itself or can be a separate file played (executed) in
15 synchronization with the image file.

Non-limiting examples of the invention's image files include, but are not limited to, corporate commercials, public service ads, current weather graphics (retrieved from the Internet or other means on a continuing basis), local and national sports team schedules, National Weather
20 Service advisories and warnings, traffic flow information, or fugitive information 3.02.

Image and audio files can be obtained from clients, user-generated, or retrieved from the Internet. Once obtained, these files are stored in an indexed database 3.03 on the data acquisition

central processing unit (or on a computer accessible via network connection to the data acquisition central processing unit).

In Figure 3, the user next creates "playlists" on the data acquisition central processing unit 3.07. These "playlists" consist of indexed references to image files, audio files, and ticker information. The "playlists" are designed and designated for groups, sub-groups, or individual remote site central processing units 3.07.

The Phase I image processing component processes these "playlists" as they are entered by the user and maintains an inventory (database) of file names located on each remote computer. When a "playlist" is created by the user, the program determines which remote site central processing units are to be included. The processing component then compares the "playlist" with the file names of the addressed remote site central processing units.

If files required to execute the "playlist" are not on a particular remote computer, then the appropriate files are transmitted directly to the remote computer or they are transmitted to a private web server 3.09. The "playlists" are also transmitted to the specified remote site central processing units.

Once a "playlist" is entered, the processing component ensures that requested files are available to the data acquisition central processing unit. If they are not, an error message is generated, and the user is prompted to download the necessary files to the data acquisition central processing unit (or to a computer residing on the same network as the data acquisition central processing unit).

The Phase I image processing component also contains sub-routines designed to detect problematic occurrences at the transmitting site. These occurrences may be, but are not limited to,

a faulty Internet connection, a loss of feedback from a remote computer, or a failure of one of the other programs running on the computer.

When problems are detected the program can send notification of these occurrences to an appropriate service technician. These notifications can be in the form of, but are not limited to, visual screen notification, e-mails, a transmission to a private web server, or a signal sent to a pager. The data acquisition central processing unit is designed with several redundant systems. First and foremost is a fully-functional computer which contains a program to verify the functionality of the data acquisition central processing unit. If any problems are detected, a secondary computer will begin operations and take over all of the ticker and image processing component transmission functions.

Several redundancies are in place for the private web server. Should the server fail for any reason, another private web server will be automatically chosen and utilized in its stead.

Figure 4 is a flow chart representing Phase II software functionality. The remote site central processing units are currently operating in a Windows® environment, but the programming is not platform specific. Phase II software source code is immediately provided to facilitate purposes of full and enabling invention disclosure.

'This form will not be visible. It will be running on the remote computer
'at the store locations

Private Sub Receive()
'Open Internet Web Site and Download New Data for Ticker
'Open Internet Web Site and Download and Changes to Video Segments
'Open Satellite Files from Satellite Receiver (both data and video)

End Sub

Private Sub Transmit()
'Transmit video feed to monitors on pumps.
'Transmit ticker to bottom portion of monitors on pumps.

'Ticker information will come from the data stream
'and will contain national news, sports scores, dow jones

'industrial avg., nasdaq avg., s&p500 avg., local news,
'local weather
End Sub

```
5  VERSION 5.00
    Begin VB.Form frmDataEnv
        Caption      = "MediaVision Internet and Satellite Receiving"
        ClientHeight  = 3195
        ClientLeft    = 60
10     ClientTop     = 345
        ClientWidth   = 4680
        LinkTopic     = "Form1"
        ScaleHeight   = 11115
        ScaleWidth    = 15240
15     StartUpPosition = 3 'Windows Default
    End

    VERSION 5.00
    Begin {C0E45035-5775-11D0-B388-00A0C9055D8E} DataEnvironment1
20         ClientHeight  = 5055
        ClientLeft      = 1080
        ClientTop       = 1500
        ClientWidth     = 6300
        _ExtentX        = 11113
25         _ExtentY     = 8916
        FolderFlags     = 3
        TypeLibGuid     = "{A5DC9AF5-9235-11D1-B067-00DD01144174}"
        TypeInfoGuid    = "{A5DC9AF6-9235-11D1-B067-00DD01144174}"
        TypeInfoCookie  = 0
30         Version      = 4
        NumConnections  = 1
        BeginProperty Connection1
            ConnectionName = "Connection1"
            ConnDispId    = 1001
35             SourceOfData = 3
            QuoteChar     = 34
            SeparatorChar = 46
        EndProperty
        NumRecordsets   = 0
40     End

    VERSION 5.00
    Begin {78E93846-85FD-11D0-8487-00A0C90DC8A9} DataReport1
45         Caption      = "DataReport1"
        ClientHeight    = 8235
        ClientLeft      = 1650
        ClientTop       = 1545
        ClientWidth     = 6585
        _ExtentX        = 11615
50         _ExtentY     = 14526
        _Version        = 393216
        _DesignerVersion= 100684101
        BeginProperty Font {0BE35203-8F91-11CE-9DE3-00AA004BB851}
            Name        = "Arial"
55             Size     = 8.25
```

```

    Charset      = 0
    Weight       = 400
    Underline    = 0 'False
    Italic       = 0 'False
5    Strikethrough = 0 'False
    EndProperty
    GridX        = 10
    GridY        = 10
    LeftMargin   = 1440
10   RightMargin = 1440
    TopMargin    = 1440
    BottomMargin = 1440
    NumSections  = 3
    SectionCode0 = 2
15   BeginProperty Section0 {1C13A8E0-A0B6-11D0-848E-00A0C90DC8A9}
        _Version    = 393216
        Name        = "Section2"
        Object.Height = 360
        NumControls  = 0
20   EndProperty
    SectionCode1 = 4
    BeginProperty Section1 {1C13A8E0-A0B6-11D0-848E-00A0C90DC8A9}
        _Version    = 393216
        Name        = "Section1"
25   Object.Height = 1440
        NumControls  = 0
    EndProperty
    SectionCode2 = 7
    BeginProperty Section2 {1C13A8E0-A0B6-11D0-848E-00A0C90DC8A9}
        _Version    = 393216
        Name        = "Section3"
30   Object.Height = 360
        NumControls  = 0
    EndProperty
35   End
End

```

'sending a video file and ticker to monitors

```

Private Sub MMControll GotFocus()
40   'MMControll.Command = 'Open"
    'MMControll.PlayVisible = True
    'MMControll.PlayEnabled = True
    'MMControll.Command = "Play"
    'MMControll.Visible = False
45   End Sub

Private Sub MediaPlayer1_DVDNotify(ByVal EventCode As Long, ByVal EventParam1
As Long, ByVal EventParam2 As Long)
50   'This will play a video file taking up the majority of the screen as defined on the form
    MediaPlayer1.FileName = "c:\windows\desktop\videol.mpg"
    MediaPlayer1.Open
    MediaPlayer1.Stop
55   End Sub

```

```

Private Sub Ticker()
'Open data file containing raw text for ticker
'format data
'send to monitors
End Sub

```

VERSION 5.00

Object = "(22D6F304-B0F6-11D0-94AB-0080C74C7E95)#1.0#0";"MSDXM.OCX"

Begin VB.Form Form1

```

Caption = "Form1"
ClientHeight = 3195
ClientLeft = 60
ClientTop = 345
ClientWidth = 4680
LinkTopic = "Form1"
ScaleHeight = 11115
ScaleWidth = 15240
StartupPosition = 3 'Windows Default
WindowState = 2 'Maximized

```

Begin MediaPlayerCtl.MediaPlayer MediaPlayer1

```

Height = 10575
Left = 240
TabIndex = 0
Top = 120
Width = 14895
AudioStream = -1
AutoSize = 0 'False
AutoStart = -1 'True
AnimationAtStart = -1 'True
AllowScan = -1 'True
AllowChangeDisplaySize = -1 'True
AutoRewind = 0 'False
Balance = 0
BaseURL = ""
BufferingTime = 5
CaptioningID = ""
ClickToPlay = -1 'True
CursorType = 0
CurrentPosition = -1
CurrentMarker = 0
DefaultFrame = ""
DisplayBackColor = 0
DisplayForeColor = 16777215
DisplayMode = 0
DisplaySize = 3
Enabled = -1 'True
EnableContextMenu = -1 'True
EnablePositionControls = -1 'True
EnableFullScreenControls = 0 'False
EnableTracker = -1 'True
Filename = "c:\windows\desktop\video 6.mpg"
InvokeURLs = -1 'True
Language = -1
Mute = 0 'False
PlayCount = 999

```

```

PreviewMode      =      0 'False
Rate             =      1
SAMILang         =      ""
SAMISyle         =      ""
5 SAMIFileName    =      ""
SelectionStart   =      -1
SelectionEnd      =      -1
SendOpenStateChangeEvents =      -1 'True
SendWarningEvents =      -1 'True
10 SendErrorEvents =      -1 'True
SendKeyboardEvents =      0 'False
SendMouseClickedEvents =      0 'False
SendMouseMoveEvents =      0 'False
SendPlayStateChangeEvents =      -1 'True
15 ShowCaptioning = 0 'False
ShowControls     =      0 'False
ShowAudioControls =      -1 'True
ShowDisplay      =      0 'False
ShowGotoBar      =      0 'False
20 ShowPositionControls =      -1 'True
ShowStatusBar    =      0 'False
ShowTracker      =      -1 'True
TransparentAtStart =      0 'False
VideoBorderWidth =      0
25 VideoBorderColor =      0
VideoBorder3D    =      0 'False
Volume           =      -600
WindowlessVideo  =      0 'False
End
30 End

```

The remote site central processing units can utilize a dial-up connection or a high-speed Internet connection. The computer can communicate directly with the data acquisition central processing unit via modem or VPN. It can also download information via FTP, HTTP, or HTTPS protocols from the private web server on the Internet.

The remote site central processing units download the "playlists" designated for their receipt 4.02. If a direct link is established with the data acquisition central processing unit, then the necessary files (image, audio, ticker information) are also received 4.03.

If there is no direct link with the data acquisition central processing unit, then the "playlist" is analyzed by the program. If the "playlist" refers to files which the remote computer does not have on its hard disk, then the files are downloaded from the private web server.

The ticker information is downloaded 4.06 or received very frequently (usually every five to ten minutes). This facilitates the timely refreshing of news and other time-critical information to the viewer.

All downloaded files are stored on the remote computer's hard drive 4.08. Audio and video files are archived when not being used, but the ticker information is usually overwritten as newer information is obtained.

The computer combines the ticker information with the image and audio files according to the "playlist". A video and audio signal are created with this combined information. This signal is sent to video monitors via a hard wire or RF signal as indicated in the hardware description 4.10.

The ticker information is normally contained within a horizontal strip at the bottom of the monitor's viewable screen area. However, the ticker can be placed at the top or in a vertical alignment. The remainder of the screen is filled with the appropriate video signal corresponding to the "playlist".

The program also contains sub-routines designed to detect problematic occurrences at the remote site 4.19. These occurrences may be, but are not limited to, a faulty Internet connection, a loss of video or audio at a video screen, or a failure of one of the other programs running on the computer.

When problems are detected the program can send notification of these occurrences to the appropriate service technician 4.20. These notifications can be in the form of, but are not limited to, e-mails, a transmission to a private web server, or a signal sent to a pager.

In some instances, the screens may have sensors located on or near them. These sensors include, but are not limited to, sonic, motion, or light. The sensors are software-controlled via a communications port hub 4.23.

The sonic sensor measures the decibel level at each screen location. This information is used by the program to control the audio volume which is sent to that screen 4.25.

The motion sensor detects when a vehicle (or a user-defined object) has approached the screen 4.27. This information is used by the program to determine when the screen is active or passive. The light sensor measures the radiance level at each screen location. This information is used by the program to control the brightness and contrast of each screen 4.29.

The program is designed to allow for input from the viewer 4.32 (or user in this instance) at each screen location. This input can be from, but is not limited to, a remote keypad, a voice recognition system facilitated by a microphone, a receiver designed to detect electronic signatures, an optical scanner, or a barcode reader.

The keypad can be numeric or alpha-numeric 4.35. The data received by the remote computer from the keypad can be utilized to respond to user queries, or to accumulate and display information based upon user preferences.

A voice recognition system can be composed of a wide-area microphone at the screen location. The microphone signal would be processed by a sound board or a specialized speech-recognition card located within the computer. This signal is interpreted via a combination of hardware and software technology. The data received by the remote computer from the voice recognition technology can be utilized to respond to user queries, or to recognize voice signatures which correspond to customizable user profiles.

An electronic signature can be detected by a receiver located at each screen location. These electronic signatures can be generated by an RF device, or other user-specific electronic transmitting mechanism. The electronic signature would be unique to each user and would identify that user to

the remote computer. Once identified, commercials or other information can be tailored specifically for that individual based upon their profile.

An optical scanner or a barcode reader can be utilized to identify specific users and customize information based upon their profile. An optical scanner or a barcode reader can also be used for the purpose of reading coupons, "tickets", or other company-generated written instruments. For example, a contest could be created and "tickets" printed up. The "ticket" could be identified by the optical scanner or the barcode reader, and the user notified as to whether or not they are a winner. Video and video/audio cameras can be located at each screen or facing towards each screen 4.37. The signals from each camera can be wired through a video control box first or they can be wired directly into a multi-function digital capture card in the computer. The signals are segregated so that the program is aware of each individual video signal.

These video/audio captures can be either time-lapsed or full-motion (up to 30 frames per second, or fps). The captured digital images and sound can be stored locally on the remote computer, streamed to another computer, or they can be transmitted to a private web server for later retrieval by an employee or designated client 4.39.

The video and audio can be used for many purposes, including but not limited to, security, contests, or verification of screen functionality.

When the audio/video is utilized as a security mechanism, the uses are many-fold. The video/audio can be used to identify vehicles which drive away from the pump without paying. It can be used to identify individuals who commit vandalism. It can be used to identify robbery suspects, or persons attempting to harm the employees of a client.

The audio/video digital recording can also be used to identify vehicle tag numbers, which can later be used to pick out the winner of a contest. This recording of vehicle tag numbers is also an effective method of locating those who drive away from the pump without paying (driveaways). A very important aspect of the video/audio recording is the ability to detect and trouble-shoot any problems with the screens. These cameras ensure that the screens are projecting an image when they should be. If there is a problem, it can be easily determined which screens are malfunctioning so that a service technician can be dispatched to make any needed hardware repairs.

Finally, the cameras at or near the screens act as a deterrent to unlawful behavior. They reduce the incidence of driveaways, and possibly even the number of robbery and vandalism attempts at a client's location.

Figure 5 illustrates high level Phase I software processes executing within the data acquisition central processing unit, accessing and collecting via a network communication means a diverse plurality of geographically relevant information. Said collated information is then displayed on a computer resident and accessible storage medium, such as a web page. Accompanying video and still images are provided as further records and also displayed on said accessible storage medium. This information is then combined and is communicated via a computer recognizable and compatible communication means via a transfer protocol such as FTP to a transceiver facility for subsequent processing by Phase II components of the instant invention. Phase II components of the information are disclosed in association with Figure 6.

Figure 6 illustrates Phase II of the instant invention wherein the combined ticker, video, still image file is transmitted to multiple geographically dispersed remote receiving locations. As each segment (a.k.a. "record") of ticker, video and still image records contain delineation control code

inserted via first software processing, only codes matching those of geographically relevant locations within a geographically dispersed area will receive and be able to decode such information. Consequently, only information for an individual remote station, group of stations, located within a neighborhood, city, town, country, etc. will receive coded information. Such coded information is then processed by software resident at the store location 6.5 and further transmitted via either RF signal or hard wire to screens 6.10 or information presentation devices 6.15 approximate to fuel pump gassing apparatuses. Such features of the display apparatuses are shown in conjunction with Figure 7.

In Figure 7 a display device 7.01 is presented. The screen display area is broken into two general areas. A display area for photographs or images 7.15 and a display area for textural images 7.20, 7.10 designates speaker areas for audio content accompanying said display images or files.

While this invention has been described to illustrative embodiments, this description is not to be construed in a limiting sense. Various modifications and combinations of the illustrative embodiments as well as other embodiments will be apparent to those skilled in the art upon referencing this disclosure. It is therefore intended that this disclosure encompass any such modifications or embodiments.

Alternate Embodiments

The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be apparent to one skilled in the art that the specific details are not required in order to practice the invention. In other instances, well known circuits and devices are shown in block diagram form in order to avoid unnecessary distraction from

the underlying invention. Thus, the foregoing descriptions of specific embodiments of the present invention are presented for purposes of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed, obviously many modifications and variations are possible in view of the above teachings. The embodiments were chosen and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents.

Further, the method and system described herein above is amenable for execution on various types of executable mediums others than a memory device such as a random access memory. Other types of executable mediums can be used, such as but not limited to, a computer readable storage medium which can be any memory device, compact disc, or floppy disk.